Local E-optimality Conditions for Trajectory Design to Estimate Parameters in Nonlinear Systems

Andrew D. Wilson and Todd D. Murphey

Abstract—This paper develops an optimization method to synthesize trajectories for use in the identification of system parameters. Using widely studied techniques to compute Fisher information based on observations of nonlinear dynamical systems, an infinite-dimensional, projection-based optimization algorithm is formulated to optimize the system trajectory using eigenvalues of the Fisher information matrix as the cost metric. An example of a cart-pendulum simulation demonstrates a significant increase in the Fisher information using the optimized trajectory with decreased parameter variances shown through Monte-Carlo tests and computation of the Cramer-Rao lower bound.

I. INTRODUCTION

The design of trajectories for the experimental identification of parameters in dynamic systems is an important problem in a variety of fields ranging from robotics to biology and beyond. With more accurate models, the performance and tuning of controllers can be radically improved. For nonlinear dynamical systems, the trajectory is constrained to a nonlinear set of equations of motion which leads to challenges in parameter estimation. Additionally, since the trajectories evolve on a continuous-time domain, it is important to ensure that the dynamics are satisfied along the entire time domain of any synthesized trajectory.

A variety of estimation techniques are used in practice, including Kalman filtering, maximum likelihood estimators, and Monte-Carlo estimators to name a few [1-3]. The focus of this paper will lie solely in the area of estimating static model parameters in nonlinear dynamical systems. A widely used method of estimating these types of parameters is through a maximum likelihood estimation technique known as batch least-squares estimation [3].

The batch least-squares method compares a set of measurements taken along the evolution of a trajectory to predicted observations of the system using the model equations and estimates of the parameters. This comparison is made using the method of least-squares along the trajectory, and a new update to the parameter estimate is then calculated.

II. RELATED WORK

The design of the experimental trajectory is of particular importance to maximize the amount of information gained from the parameter estimation task. A large amount of literature on optimal experimental design exists in the fields of biology [4-6], chemistry [7], and systems [8-11].

A common metric used in the area of experimental design that will also be a key metric in this paper is the Fisher information matrix computed from observations of the system trajectory [12]. Metrics based on Fisher information are used as cost functions in many optimization problems including work by Swevers on “exciting” trajectories [13]. This work, as well as related works [14], [15], synthesize trajectories for nonlinear systems that can be recast as linear systems with respect to the parameters.

Further research has resulted in optimal design methods for general nonlinear systems. In work by Emery [16], similar least-squares and maximum likelihood estimation techniques are combined with Fisher information to optimize the dynamic experiment. In this case and a number of others, the dynamics are solved as a discretized, constrained optimization problem [17], [18].

To avoid a discretization of the dynamics of the continuous system, a class of methods has been developed which relies on sets of basis functions to synthesize an optimal control input to the system [19-21]. This method allows the full trajectory to be optimized on a continuous-time domain; however, the optimization is still subject to a finite set of basis function coefficients.

The main contribution of this paper is the formulation of an infinite-dimensional, projection-based optimization algorithm which maximizes the information gained from observations along the dynamic trajectory. The projection-based algorithm was originally designed for optimal trajectory tracking problems [22], [23]; however, this paper extends the algorithm to include a non-Bolza cost function, maximizing the information gained by observations of the dynamic trajectory. This formulation results in a method that preserves the continuous-time dynamics using variational perturbations to the input and trajectory to find an optimal solution.

This paper is organized as follows: Section III introduces the estimation concepts required to compute the Fisher information matrix for the system. Section IV derives the required cost function and equations necessary to optimize the trajectory over the Fisher information. Finally, an example of the optimization algorithm is provided for a cart-pendulum system in simulation.

III. LEAST-SQUARES ESTIMATION

This paper will assume that a predefined number of model parameters are being estimated from a series of state measurements taken along a dynamic system’s trajectory evolution. To provide background for the trajectory optimization objective function, which will be presented in the next
section, the estimation procedure will first be outlined. The dynamic model of the system is defined as 
\[ \dot{x}(t) = f(x(t), u(t), \theta) \] (1)
where \( x \in \mathbb{R}^n \) defines the system states, \( u \in \mathbb{R}^m \) defines the inputs to the system, and \( \theta \in \mathbb{R}^p \) defines the set of model parameters that will be estimated. For the purpose of this paper, all states will be assumed to be observable for the estimation algorithm.

When running an experiment, a predefined trajectory, \( x(t) \), is executed, and measurements of the state are taken during the evolution of the experiment. We will assume that the trajectory has a finite time horizon, and a fixed number of observations are made. This assumption commonly occurs for fixed frequency sampling of sensors over a fixed time horizon in an experimental setup. Using the collected measurements, least-squares estimation of the parameter set can be performed using a Newton-Raphson search method. The objective of the least-squares estimator can be rewritten as 
\[ \hat{\theta} = \arg \min_{\theta} \lambda(\theta) \]
where 
\[ \lambda(\theta) = \sum_{i} (\tilde{x}(t_i) - x(t_i))^T \Sigma^{-1} \cdot (\tilde{x}(t_i) - x(t_i)). \]
\( \tilde{x}(t_i) \) is the observed state at the \( i^{th} \) time-point of \( h \) measurements and \( \Sigma \in \mathbb{R}^{n \times n} \) is the covariance matrix associated with the sensor measurement error.

Given this minimization problem, the Newton-Raphson iterator is given by 
\[ \hat{\theta}_{k+1} = \hat{\theta}_k - [\nabla^2 \lambda(\theta)]^{-1} \nabla \lambda(\theta). \]

A. Fisher Information

The Fisher information matrix quantifies the amount of information a set of observations contains about a set of unknown parameters. Assuming that the measurement noise of the system is normally distributed, the Fisher information matrix for the linear estimator is given by 
\[ I(\theta) = \sum_{i} \nabla \theta x(t_i) \cdot \Sigma^{-1} \cdot \nabla \theta x(t_i)^T. \]

Since \( I(\theta) \in \mathbb{R}^{n \times n} \), an appropriate metric must be chosen which involves the Fisher information matrix. There is a significant amount of literature on different types of mapping choices [4], [16], [17]; however, this paper will restrict itself to the design choice of E-optimality. This choice of mapping attempts to improve the worst-case variances of the parameter set by maximizing the minimum eigenvalue of the Fisher information matrix.

Given this choice of mapping, the objective function for the trajectory optimization includes the inverse of the minimum eigenvalue as well as cost on the control effort. Additionally, a trajectory cost may be added to keep optimal trajectories in the neighborhood of the initial trajectory. The details of the trajectory optimization cost and routine will be presented in the following section.

IV. Nonlinear Trajectory Optimization

A. Objective Function

The trajectory optimization problem described in the previous section will now be formally defined. The objective function, \( J \) is dependent on the eigenvalues of the Fisher information matrix, \( I(\theta) \). Since our optimization method requires a continuous cost function, the maximization of the information matrix needs to be cast into an appropriate continuous analogue. To satisfy this condition, the information equation will be written as 
\[ \tilde{I}(\theta) = \int_{t_0}^{t_f} \nabla_\theta x(t) \cdot \Sigma^{-1} \cdot \nabla_\theta x(t)^T dt. \] (2)
Assuming that observations are taken regularly along the entire trajectory, the optimal trajectory \( x^*(t) \) maximizing the eigenvalues of the continuous \( \tilde{I}(\theta) \) will approximately optimize the eigenvalues of the sampled \( I(\theta) \). As the sampling rate increases, the values of \( \tilde{I}(\theta) \) and \( I(\theta) \) will converge.

If observations do not take place along the entire trajectory and the times samples are measured along the trajectory are predetermined, a weighting function can be added to \( \tilde{I}(\theta) \) to ensure that sensitivity is maximized in the sampled areas of the trajectory. However, if the observation time is not predetermined, the sensitivity along the entire trajectory will be maximized using \( \tilde{I}(\theta) \).

The optimization objective function will therefore be given by 
\[ J = \frac{Q_p}{\lambda_{\min}} + \frac{1}{2} \int_{t_0}^{t_f} [(x(t) - x_d(t))^T \cdot Q_T \cdot (x(t) - x_d(t)) + u(t)^T \cdot R_T \cdot u(t)] dt \] (3)
where \( \lambda_{\min} \) is the minimum eigenvalue of \( \tilde{I}(\theta) \), \( Q_p \) is the information weight, \( x_d(t) \) is a reference trajectory, \( Q_T \) is a trajectory tracking weighting matrix, and \( R_T \) is a control effort weighting matrix.

The various weights allow for design choices in the optimal trajectory that is obtained. The control weight is required to maintain a convex optimization problem and must be a positive definite matrix. Increasing this weight will result in less aggressive trajectories, most likely decreasing the obtained information. Using a reference trajectory allows for an optimal solution that remains in the neighborhood of a known trajectory.

To minimize the objective function given by (3), an optimal control algorithm is formulated using an infinite dimensional, projection-based approach. The technique is founded on the approach detailed in [22], but is extended in this paper to allow for a cost function in the form of (3).

B. Extended Dynamics Constraints

For the optimization of the system trajectory, the objective function must be minimized while satisfying the governing dynamic equations of the system. The nonlinear system dynamics are given by (1).
The optimal control algorithm traditionally was formulated for trajectory tracking problems where the objective function is explicitly a function of the system states. However, given the formulation of the objective function in the previous section, the cost depends on \( \nabla_\theta x(t) \). The Jacobian along the trajectory can be calculated by first writing \( x(t) \) in its integral form,

\[
x(t) = x_0 + \int_{t_0}^{t_f} f(x(t), u(t), \theta) dt.
\]

Calculating the Jacobian with respect to \( \theta \), yields

\[
\nabla_\theta x(t) = \int_{t_0}^{t_f} D_x f(x(t), u(t), \theta)^T \nabla_\theta x(t) + D_\theta f(x(t), u(t), \theta)^T dt.
\]

If the above equation is now differentiated with respect to \( \psi \) for variations on \( x \) as an additional dynamic constraint. This will allow variations of the trajectory to be calculated free of the constraint of maintaining feasible dynamics; however, the solution is projected to a feasible trajectory at each iteration of the optimization algorithm.

### D. Optimization Routine

The optimal control problem is solved using a gradient descent technique. In order to apply the iterative method, a descent direction must be defined for each iteration of the algorithm. The descent direction, \( \zeta_i(t) \) is given by

\[
\zeta_i(t) = \arg \min_{\zeta} DJ(P(\xi_i(t))) \circ \zeta_i(t) + \frac{1}{2} \langle \zeta_i(t), \zeta_i(t) \rangle
\]

where \( \zeta_i(t) \in T_{\xi_i T} \), i.e., the descent direction lies in the tangent space of the trajectory manifold at the current iteration. The components of the descent direction are given by \( \zeta_i = (\bar{z}(t), v(t)) \) where \( \bar{z}(t) \) is the perturbation to the extended state and \( v(t) \) is the perturbation to the control.

The solution to (5) results in a descent direction for the optimal control problem (3) and an appropriate Armijo step of the projection, \( P(\xi_i(t) + \gamma_i \zeta_i) \) provides a feasible trajectory solution assuming that the step size \( \gamma_i \) satisfies a sufficient decrease condition.

The basic algorithm is shown in Algorithm 1. The descent direction is found by defining a local LQR optimization problem at each iteration of the trajectory [24]. The following section will cover the setup and structure of this LQR problem. Sufficient decrease is satisfied by an Armijo line search on the solution which is projected onto the trajectory manifold [25]. After the steepest descent condition is satisfied, the updated trajectory becomes the seed for the next iteration of the optimal control algorithm.

#### Algorithm 1 Trajectory Optimization

- Initialize \( \xi_0 \in T \), tolerance \( \epsilon \)
- while \( DJ(\zeta(t)) \circ \zeta > \epsilon \) do
  - Calculate descent, \( \zeta_i \):
    \[
    \zeta_i = \arg \min_{\zeta_i(t)} DJ(P(\xi_i(t))) \circ \zeta_i + \frac{1}{2} \langle \zeta_i(t), \zeta_i(t) \rangle
    \]
  - Compute \( \gamma_i \) with Armijo backtracking search
  - Project onto dynamics constraints:
    \[
    \xi_{i+1}(t) = P(\xi_i(t) + \gamma_i \zeta_i)
    \]
  - \( i = i + 1 \)
- end while

### E. Solving the LQR Problem

To find a descent direction for the optimal control algorithm, the LQR problem must first be formulated. As shown in (5), the descent direction depends on the linearization of the cost function, \( DJ(P(\zeta_i(t))) \) and the local quadratic model, \( \frac{1}{2} \langle \zeta_i(t), \zeta_i(t) \rangle \). The following subsections present the formulations for these two quantities as well as the linearization of the dynamics which constrain the descent direction.
The derivative of the outer product is straightforward. The matrix
\( \frac{\partial f}{\partial x} = \frac{1}{\lambda_{\min}} \frac{\partial \lambda_{\min}}{\partial x} \) 
+ \int_{t_0}^{t_f} [(x(t) - x_d(t))^T \cdot Q \cdot x(t)] \, dt. \tag{6}

Since the cost function involves the eigenvalues of \( \hat{I}(\theta) \),
this linearization requires a method of handling derivatives
of eigenvalues. An equation for derivatives of eigenvalues
was formalized by Nelson \[26\]. Given an eigensystem of
the form
\( AX = \lambda X \)
where \( \lambda \) is a diagonal matrix of eigenvalues, \( \{\lambda_1, \lambda_2, ..., \lambda_n\} \),
and \( X \) is the associated matrix of eigenvectors, the derivative
of one eigenvalue is given by
\( D_x \lambda_k = y_k^T \cdot D_x A \cdot v_k \)
where \( y_k \) is the left eigenvector and \( v_k \) is the right eigenvector
associated with \( \lambda_k \).

Using this result, \( \frac{\partial \lambda_k}{\partial x} \) from (6) can be calculated. Taking
the derivative of the eigenvalue of \( \hat{I}(\theta) \) from (2) with respect
to the extended state yields
\( \frac{\partial \lambda_i}{\partial \hat{x}} = y_i^T \frac{\partial}{\partial x} \left( \int_{t_0}^{t_f} \nabla_{\theta \hat{x}(t)} \cdot \Sigma^{-1} \cdot \nabla_{\theta \hat{x}(t)}^T \, dt \right) v_i \)
where \( i \) denotes the index of the minimum eigenvalue and
eigenvector. Since the partial derivative and eigenvectors
are not time dependent, the equation can be rewritten as the
running cost over the derivative of the outer product of the
Jacobians,
\( \frac{\partial \lambda_i}{\partial \hat{x}} = \int_{t_0}^{t_f} y_i^T \frac{\partial}{\partial x} \left( \nabla_{\theta \hat{x}(t)} \cdot \Sigma^{-1} \cdot \nabla_{\theta \hat{x}(t)}^T \right) v_i \, dt. \)

The last step in computing the extended state linearization
of the cost function is calculating the derivative of the outer
product of the Jacobians, \( \frac{\partial \nabla_{\theta \hat{x}(t)} \cdot \Sigma^{-1} \cdot \nabla_{\theta \hat{x}(t)}^T}{\partial x} \). Given that
\( \nabla_{\theta \hat{x}(t)} \) is included in the extended state, \( \bar{x}(t) \), the partial
derivative of the outer product is straightforward. The matrix
representation of the derivative is given by
\( \frac{\partial}{\partial x} \left( \nabla_{\theta \hat{x}(t)} \cdot \Sigma^{-1} \cdot \nabla_{\theta \hat{x}(t)}^T \right) = \left[ \begin{array}{c} \{0\} \\
2 \nabla_{\theta \hat{x}(t)} \cdot \Sigma^{-1} \end{array} \right] \)
where \( E \) is an identity-like tensor of the form
\( E_{i,j,k,l} = \delta_{i,k} \delta_{j,l} \)
with \( \delta_{.} \) as the Kronecker delta function.

Given the cost function (3), the linearization has been
defined in terms of the expanded state, \( \bar{x}(t) \). Additionally, the
linearization with respect to the control \( u(t) \) is needed. \( \frac{\partial f}{\partial x} \)
is much simpler since only the control cost term is directly
dependent on the control. Therefore, the linearization is given by
\( b(t) = \frac{\partial J}{\partial u} = \int_{t_0}^{t_f} [u(t)^T \cdot R \cdot x(t)] \, dt. \tag{7} \)

These linearization terms are evaluated using the state and
control at each point in time along the trajectory.

2) Quadratic Model: The second required component of
the LQR problem is defining the inner product, \( \frac{1}{2} \langle \zeta_i, \zeta_i \rangle \). For
this paper, we will choose a simple quadratic model related
to the extended state and control inputs. This will be defined as
\( \frac{1}{2} \langle \zeta_i, \zeta_i \rangle = \int_{t_0}^{t_f} \frac{1}{2} \bar{z}(t)^T Q(t) \bar{z}(t) + v(t)^T R(t) v(t) \, dt \)
where matrices \( Q(t) \) and \( R(t) \) are weighting matrices for
the local quadratic model approximation. Design of these
weighting matrices can lead to faster convergence of the
optimal control algorithm depending on the specific problem.

3) Dynamics Linearization: The final step required to set
up the LQR problem is to calculate linearizations of the
dynamics. Since the original dynamics for the system are
nonlinear, linearized dynamics are used as an approximation
for the steepest descent calculation and then the projection
operator projects the step back onto the nonlinear dynamics.
The descent direction, \( \zeta(t) \), will satisfy the linear constraint
ODE given by
\( \dot{\zeta}(t) = A(t) \zeta(t) + B(t) v(t) \)
where \( A(t) \) is the linearization of the nonlinear dynamics
given by (1) and (4) with respect to \( \bar{x}(t) \), and \( B(t) \) is the
linearization with respect to \( u(t) \). The linearization, \( A(t) \),
of the dynamics with respect to the extended state, \( \bar{x}(t) \), is
given by
\( A(t) = \left[ \begin{array}{c}
\frac{\partial \hat{x}}{\partial x} \\
\frac{\partial \hat{\theta}}{\partial x} \\
\frac{\partial \hat{\phi}}{\partial x} \\
\frac{\partial \hat{\phi}}{\partial \theta}
\end{array} \right] \)
\( = \left[ \begin{array}{c}
D_x f(\cdot) \\
D_{\theta} f(\cdot) \cdot \psi(t) + D_x D_{\theta} f(\cdot) \{0\} \\
D_{\phi} f(\cdot) \cdot \psi(t) + D_x D_{\phi} f(\cdot) \end{array} \right]. \)

Additionally, the linearization of the dynamics with respect
to the control input, \( u(t) \), is required. This linearization
matrix, \( B(t) \), is given by
\( B(t) = \left[ \begin{array}{c}
\frac{\partial \hat{x}}{\partial u} \\
\frac{\partial \hat{\theta}}{\partial u} \\
\frac{\partial \hat{\phi}}{\partial u}
\end{array} \right] \)
\( = \left[ \begin{array}{c}
D_u f(\cdot) \\
D_{\theta} f(\cdot) \cdot \psi(t) + D_u D_{\theta} f(\cdot) \\
D_{\phi} f(\cdot) \cdot \psi(t) + D_u D_{\phi} f(\cdot)
\end{array} \right]. \)

Given the cost function linearization and quadratic model,
the LQR problem which solves for the descent direction (5)
can be written as
\( \arg \min_{\zeta(t)} \int_{t_0}^{t_f} a(t)^T \zeta(t) + b(t)^T v(t) \)
\( + \frac{1}{2} \zeta(t)^T Q(t) \zeta(t) + \frac{1}{2} v(t)^T R(t) v(t) \, dt \tag{8} \)
such that
\[ \dot{z}(t) = A(t)\dot{z}(t) + B(t)v(t) \]
where \( a(t) = \frac{\partial a}{\partial s} \), \( b(t) = \frac{\partial a}{\partial s} \), and \( A(t) \) and \( B(t) \) are the linearizations of the system dynamics. The problem is therefore reduced to solving this LQR problem iteratively to find directions of steepest descent. The solution to the LQR descent problem can then be solved using the known Riccati differential equations \([22]\). Since this descent direction is based on the linearized dynamics, the projection operator must be applied to ensure the dynamics constraints are satisfied. This process is iteratively repeated until convergence is achieved as shown in Algorithm 1.

V. Simulation Example

To demonstrate the procedure and use of optimizing the Fisher information for a dynamic system, a simulation of a cart-pendulum system is considered. The system has two degrees of freedom, \( (s(t), \phi(t)) \), where \( s(t) \) is the horizontal displacement of the cart and \( \phi(t) \) is the rotational angle of the pendulum as seen in Fig. 1.

\[ \frac{d}{ds} - \frac{d}{dt} \frac{\partial L}{\partial \dot{s}} = u(t); \quad \frac{\partial L}{\partial \phi} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} = -c\dot{\phi}(t), \]

where \( M \) is the cart mass, \( m \) is the pendulum mass, \( \ell \) is the pendulum length, and \( c \) is the viscous friction parameter. The full equations will not be derived here; however, the results are easily obtained.

A. Simulation Parameters

The goal for this trajectory optimization simulation will be to estimate three parameters given noisy measurements of the full state of the system. The three parameters will be the mass of the pendulum, \( m \), the length of the pendulum, \( \ell \), and the friction coefficient, \( c \).

An experimental measurement of the trajectory is simulated by sampling the trajectory of the system given the current estimate of parameters at a discrete number of points. Additive noise is then added to each sample which results in a simulated set of noisy measurements that will be used for the subsequent optimization routine. For this simulation example, a 4 second trajectory will be measured with a fixed sampling rate of 100 Hz. The uncertainty of each state measurement will be normally distributed with zero mean and a standard deviation, \( \sigma = 0.05 \).

An initial control input is chosen as shown in Fig. 2c. An initial guess of the parameters, \( \theta = \{m, \ell, c\}, \) is chosen as well as the actual deterministic parameter set. The parameters and estimates used to initialize the optimization are:

- Actual: \( m = 2.0 \, \text{kg}, \, \ell = 1.0 \, \text{m}, \, c = 1.0 \, \text{Ns/m} \)
- Guess: \( m = 2.5 \, \text{kg}, \, \ell = 1.4 \, \text{m}, \, c = 1.4 \, \text{Ns/m} \)

The cart mass will be assumed to be known and fixed at 1.0 kg.

B. Optimization Results

The optimization was run until the convergence criterion \( |DJ(\xi(t)) \circ \zeta| < 10^{-2} \) was satisfied. The comparison of initial and optimized trajectories can be seen in Fig. 2. By examining Fig. 2d, it is clear that the optimized trajectory dramatically improves the amount of Fisher information as the trajectory evolves in time. This increase in information is due to the fact that the system is being excited in a way that allows the parameters to be observed more accurately. As an intuitive example, the increased oscillation of the pendulum, seen in Fig. 2b, allows a better estimate of the friction parameter, \( c \), to be made since the system is subject to greater frictional torques.

In terms of the optimization cost, the initial and optimized eigenvalues of \( \hat{I}(\theta) \) and \( J \) are listed in Table I. The results show that the minimum eigenvalue, \( \lambda_3 \), increases by over a factor of 100. Additionally, the other eigenvalues also increase, though not included directly in the cost function. The results can also be visualized through the phase-space representation of the sensitivities, \( \psi(t) \). Fig. 3 shows the phase plot of \( \nabla_c \phi(t) \) which indicates that the optimized trajectory maximizes the Fisher information by quickly increasing the sensitivity of the system throughout the trajectory.

Examining the plots of the optimized trajectory, it is clear that more information is gained by oscillating the pendulum back and forth. In particular, more information about the torsional friction parameter, \( c \), is gained by increasing the oscillation. This observation leads to a hypothesis that the cost function and optimization may be driven strongly by information concerning the friction parameter.

This hypothesis is confirmed by examining the eigenvector associated with \( \lambda_3 \). The eigenvector is \( \{0.0060018, 0.066900, 0.99774\} \), which indicates that the largest initial direction of the eigenvalue and therefore the optimization is \( c \). This analysis validates the qualitative observations made about the optimization results; however, another important test is to confirm that the variance of the parameter estimate indeed decreases with the increased information.
Fig. 2: Comparisons of the trajectory before and after Fisher information optimization.

(a) Cart position, \( s(t) \)

(b) Pendulum angle, \( \phi(t) \)

(c) Control input, \( u(t) \)

(d) Square root of the minimum eigenvalue of the Fisher information matrix, \( \sqrt{\lambda_{\text{min}}} \), over time.

Fig. 3: Phase plot of \( \nabla_c \phi(t) \) from initial (blue) and optimized (red) trajectories. Information is maximized by a trajectory that quickly causes the sensitivity w.r.t. parameters to increase.

Fig. 4: Histogram of the distribution of estimated values of the friction parameter, \( c \), from Monte-Carlo analysis.

Table I: Optimization Cost

<table>
<thead>
<tr>
<th></th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
<th>( J )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>38.170</td>
<td>19.266</td>
<td>0.87185</td>
<td>1.1470</td>
</tr>
<tr>
<td>Optimized</td>
<td>16482.9285</td>
<td>928.51</td>
<td>101.24</td>
<td>0.0098775</td>
</tr>
</tbody>
</table>

C. Monte-Carlo Analysis

To quantitatively compare the variance of the parameter estimates for both trajectories, a Monte-Carlo test was performed with both the initial and optimized trajectories using the batch-least squares estimation method. The parameter estimation routine was run 1500 times for each of the trajectories. For each trial, a new set of sampled random noise was used.

A histogram of the distribution of estimated values of the friction parameter, \( c \) can be seen in Figure 4. The covariance matrix for all three parameters computed from the Monte-Carlo data as well as the Fisher information matrix, \( I(\theta) \) for each trajectory can be seen in Table II.

The results from the Monte-Carlo tests confirm that the Fisher information significantly increases when the optimized trajectory is used in estimating the model parameters. Additionally, this increase in information leads to a significant decrease in the variance of the parameter estimates. In particular, the variance of the friction coefficient, \( c \) decreases from 0.0595 Ns/m to 0.000170 Ns/m.

D. Cramer-Rao Lower Bound

The final comparison that will be made with the results obtained from the simulation is with regard to the Cramer-Rao bound. Since it is assumed that the measurement noise is normally distributed with zero mean and the estimation
method is least-squares, the Cramer-Rao bound on the variance simply states that

$$\text{cov}_\theta(\beta) \geq I(\theta)^{-1}$$

where $\beta$ is the batch least-squares estimator [27]. This places an absolute lower bound on the variance of the parameter estimate that can be obtained using the batch least-squares estimator or other unbiased estimator.

Table III lists the covariance bounds for the initial and optimized trajectories. The covariance of the initial trajectory is clearly subject to a higher bound than that of the optimized trajectory. Due to round-off and other numerical errors in the algorithms and Monte-Carlo simulations, the covariance of the Monte-Carlo estimates is higher than the lower bound; however, overall remains quite close to the predicted best-case variance estimates according to the Cramer-Rao bound.

VI. CONCLUSION

This paper presented a method of performing an optimization of the Fisher information subject to the dynamics of a nonlinear system. By extending an infinite-dimensional projection-based trajectory optimization algorithm, the continuous-time dynamics could be preserved throughout the optimization, and variations on the input and trajectory were projected onto the trajectory manifold, $\mathcal{T}$.

The results of the cart-pendulum simulation show that this optimization routine results in an increase in the minimum eigenvalue of the Fisher information as well as a decrease in the covariance of the estimated parameters subject to

TABLE II: Covariance and Fisher Information

<table>
<thead>
<tr>
<th></th>
<th>Monte-Carlo Covariance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1.10 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$-1.51 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$3.54 \times 10^{-4}$</td>
</tr>
<tr>
<td>Optimized:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$2.75 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$8.68 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$1.65 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fisher Information Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$9.61 \times 10^2$</td>
</tr>
<tr>
<td></td>
<td>$2.21 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>$-7.26$</td>
</tr>
<tr>
<td>Optimized:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$3.16 \times 10^3$</td>
</tr>
<tr>
<td></td>
<td>$-9.24 \times 10^4$</td>
</tr>
<tr>
<td></td>
<td>$2.45 \times 10^4$</td>
</tr>
</tbody>
</table>

Fig. 4: PDF histograms of the viscous friction coefficient, $c$.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$1.04 \times 10^{-3}$</td>
</tr>
<tr>
<td></td>
<td>$-3.01 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$2.78 \times 10^{-4}$</td>
</tr>
<tr>
<td>Optimized:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$2.63 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$8.30 \times 10^{-5}$</td>
</tr>
<tr>
<td></td>
<td>$1.51 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Fig. 5: PDF histograms of the mass parameter, $m$.

TABLE III: Cramer-Rao Lower Variance Bounds
the Cramer-Rao lower bound. Future work related to the optimization approach involves exploring bounds on the convergence rates including adding second-order information to the optimization algorithm. Additionally, experimental trials are planned to compare simulation data to physical measurements.

ACKNOWLEDGMENT

This material is based upon work supported by the National Institute of Health under NIH R01 EB011615 and by the National Science Foundation under Grant CMMI 1334609. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES